Hindawi
Mathematical Problems in Engineering
Volume 2022, Article ID 5293083, 6 pages
https://doi.org/10.1155/2022/5293083

Research Article

A High-Feasibility Secure Routing against Malicious Peer in Structured P2P

Feng Wang[1] **and Chunqing Xuan**[2]

[1]*Computer Department, Zhoukou Normal University, Zhoukou 466001, Henan, China*
[2]*Finance and Economics, Zhengzhou Chenggong University, Zhengzhou 451200, Henan, China*

Correspondence should be addressed to Feng Wang; 20091021@zknu.edu.cn

Received 5 July 2022; Accepted 11 August 2022; Published 28 September 2022

Academic Editor: Zaoli Yang

As applications based on the structured peer-to-peer network have increased, the importance of security is increasing. Routing is the core of the structured peer-to-peer network, naturally which becomes the primary target of malicious nodes. The current attacks on the routing by malicious nodes are mainly sybil attacks, eclipse attacks, and routing table poisoning. In previous studies of defending above attacks, either adding redundancy to achieve security or sacrificing network scalability for security. So we establish a mathematical model of the routing process, and through the model, we analyze sybil attacks, eclipse attacks, and routing table poisoning. The same essence is found that these attacks all undermine the original convergence of the query path, and with the convergence detection, we propose the security mechanism HFS-Routing, and we design experiments and analyze the results. The results show that HFS-Routing has a lower overhead, better scalability, and higher detection rates for the malicious nodes, which is a highly feasibility mechanism.

1. Introduction

With the maturity of structured peer-to-peer networking technology represented by DHT, the application based on the peer-to-peer network has been rapidly developed, while most applications require that the network must have higher security. Therefore, many researchers gradually study the security issues of the peer-to-peer network.

DHT mainly maps the physical <ip: port> of the participating nodes to id by the HASH algorithm and then stores the corresponding < ip: port,id> in a distributed manner according to id, thereby maintaining a specific logical structure; for example, the chord protocol maintains a logic Ring topology, Pastry protocol to maintain a logical tree topology. Resource sharing is the core service of P2P applications. Resource search efficiency has become the main measure of P2P performance. It is doubtless that search efficiency depends on routing protocols. In DHT, a mandatory convergence routing algorithm is commonly used to maintain topology. However, since the nodes can freely join the network, it is easy to undermine the convergence of the routing protocols. So the routing protocol security needs to be given priority protection.

Fujii, et al. pointed out that an attacker can easily insert a large number of malicious nodes into a structured peer-to-peer network, which reduces the performance of routing [1]. To deal with this attack, a secure routing protocol is required. H.Ismail, et al. pointed out that routing constitutes the core function of P2P, and naturally, most threats attempt to destroy the peer routing table [2]. Eichert et al. pointed out that the main goal of a routing attack is to disable the p2p network from forwarding the message to the destination node [3]. Jaideep, et al. pointed out that the main attacks on peer-to-peer networks include sybil attacks, eclipse attacks, and pollution attacks [4].

In order to defend the attacks of malicious nodes on the routing, this paper presents a highly feasible routing mechanism named HFS-Routing, which can effectively defend sybil attack, eclipse attack, pollution attack, and so on.

2. Related Work

Medina et al. proposed an SDN approach to detect targeted attacks in P2P fully connected overlays [5]. Luo proposes a secure routing protocol, Symmetric-Chord [6], which obtains multiple query results by querying both forward and reverse directions on the chord ring. Finally, whether there are malicious nodes in the path is judged according to whether the query results are consistent or not. However, the protocol requires additional query messages, which adds redundancy to the network. Ismail et al. proposed a malicious eviction mechanism (EM) for P2P overlays, which uses the divergent lookups to query the target and then analyzes whether there are malicious nodes in the query result [7]. This method also requires additional queries, increasing redundancy; in addition, divergent itself undermines the convergence of the path. Shen proposes a resilient routing table, the node by maintaining a variable-size routing table to reduce the harm of malicious nodes, and the disadvantage is which needs to maintain a large number routing table [8]. Xu proposes a routing algorithm that can bypass malicious nodes but does not consider the issue of load balancing [9]. Han et al. proposed a trust-based routing strategy to solve the security problem for structured P2P networks [10]. Quantified trust is used to select the next hop and neighbor. However, trust acquisition requires additional computing resources. Cholez et al. proposed a network crawler to detect malicious nodes, but it is only suitable for the Kademlia protocol [11]. Lu et al. proposed the P2P routing security mechanism SAP2PRMEDT based on multiple encryptions, which periodically detect malicious nodes through encryption technology, which affects network scalability [12]. Therefore, the above research either adds redundancy to achieve security or sacrifices the scalability of the network for security or is only for some specific network, and our proposed security routing mechanism HFS-Routing can effectively avoid the above problems.

3. System Model

3.1. DHT Routing Model. Routing is the process that the query message is forwarded to the correct destination node, using a route table and forwarding algorithm. Castañeda et al. pointed out that the routing depends on the query message and the data structure stored on the node. In structured peer-to-peer networks, the data structures stored on the nodes are mainly routing tables [13]. The DHT technology uses a mandatory convergence routing algorithm to store < ip: port, id > based on the node's id to construct a routing table, and the entire network forms a logic on the topology. For example, chord protocol routing algorithm identifier space as a ring, the ring is $[1,2^{i-1}]$, and i is the length of the id. Each node in the chord protocol saves not only the <ip: port, id> of the predecessor and subsequent nodes that logically (by id) but also a finger table that facilitates quick lookups. Routing process is to find the corresponding physical address < ip: port, id>. For studying, we consider the entire peer-to-peer network as graph G, any node $n \in N$ that participates in the network, if any two adjacent nodes n_i and n_j have a neighbor relationship with each other. Then, n_i and n_j form an edge e_{ij}, e_{ij} and $i \neq j$, its id, respectively, id_i,id_j, $e_{ij} = \mathrm{I}id_i\text{-}id_j\mathrm{I}$, and then,

$$G = (N, E). \tag{1}$$

Assuming that the forwarding algorithm is *forward*, the query message is *query*, and then, the message forwarding process is as follows:

$$e = \text{forward}(n, \text{query}). \tag{2}$$

Assuming that *path* is the final path of the query message, r is the routing algorithm, the routing model can be expressed as follows:

$$\text{Path} = r(N, E) = e_{xy}e_{yz} \ldots e_{ij}\, Y = f(x) + z. \tag{3}$$

It is assumed that the query message returns the path $e_{xy}e_{yz}...e_{ij}$ within the TTL. If the edge e_{ij} contains the target node n_j of the query message, then $e_{xy}e_{yz}...e_{ij}$ is a successful path. On the contrary, $e_{xy}e_{yz}...e_{ij}$ is failed path.

3.2. Degrading the Routing Mechanism by Malicious Nodes. There are three main attacks on the routing: sybil attacks, eclipse attacks, and routing table poisoning.

3.2.1. Sybil Attack. Nodes in a peer-to-peer network are free to join the network, and it is this freedom that allows malicious nodes to fake multiple identities participating in the network. These fake identities are often referred to as sybil nodes, and malicious nodes can then exploit sybil node to implement a variety of attacks. The harm of sybil attack is mainly to (i) destroy routing and (ii) destroy the integrity of stored file resources. In this paper, we confine ourselves to studying the effect on routing. When the sybil node receives the query message, according to the formula (2), sybil node breaks routing in two ways: (1) providing the wrong edge as the next hop; (2) does not provide any edge. Either way, the original convergence of the routing function r is ruined.

3.2.2. Eclipse Attack. Eclipse attack is that an attacker who want to invade a node n, so he adds enough fake-node informations to the route table of node n in order to isolate the node n from the normal P2P network. Eclipse attack is a special kind of sybil attack, so eclipse attack on the impact of the path is also undermined the convergence of the routing function r.

3.2.3. Route Table Poisoning. Routing table poisoning is to use the wrong route entries to replace the normal, and this will result in receiving the query message that cannot be transmitted according to the protocol, disturbing the normal routing mechanism, in essence, undermining the routing algorithm convergence.

From the above analysis, it can be seen that the attack of malicious nodes on routing is essentially the destruction of the convergence of the routing mechanism.

3.3. Security Mechanism HFS-Routing. According to the routing model, if the convergence of the query path is not enough, the routing will fail. On the other hand, if the query fails and there are divergent nodes in the query path, then the node is a malicious node, so we will propose a routing based on the failed path named HFS-Routing, HFS-Routing only saves failed paths information, so that it will not increase additional query messages and does not require additional bandwidth, so it has a higher feasibility and better scalability. In addition, the detection method is not related to topology, therefore which is universal and suitable for all peer-to-peer protocols.

Each node saves its recent query failed paths as shown in Figure 1, pi is the query failure path, and multiple paths form graph, and whenever the query fails, HFS-Routing updates failed paths graph and triggers a malicious node detection model FPD-Detect.

3.4. Detection Model FPD-Detect. The FPD-Detect detection mechanism mainly detects whether there is a node (it is a malicious node) that causes path divergence from the failed path tree, and when the query fails which triggers the detection.

For the convenience of description, we agree that the edge from any node x to y is e_{xy},. $x,y \in n$, $x \neq y$. If there are edges e_{xy} and e_{yz} in a failed query path, edge e_{xy} precedes edge e_{yz}, we agree that $e_{xy} \prec e_{yz}$. Next, we describe how to judge malicious nodes in the detection process.

(1) If there is $e_{xy} \prec e_{yz}$ in any failed query path, subpath $e_{xy}e_{yz}$ are not converge than other subpath, then the nonroot node x may be a malicious node. For example, in Figure 1, $e_{n2n3} \prec e_{n3n4}$ exist, subpath $e_{n2n3}e_{n3n4}$ are not converge, and then, n_2 may be a malicious node.

(2) If multiple paths pass through the same node, the path with the smallest edges converges quickly. There are malicious nodes in the path with more edges, and the first node in the nonroot node is a malicious node.

(3) If there is a leaf node in multiple paths, then the node is a malicious node.

After the malicious node is detected, the routing entry corresponding to the malicious node is deleted from the routing table.

3.4.1. Detection Algorithm FPD-Detect. On the node, we design graph G to store the failed path information and concretely use the adjacency list to realize the physical storage of G.

Pseudocode test code is as follows in Algorithm 1.

4. Experimental Analysis

In order to further evaluate the performance of HFS-Routing, we will generate 10,000 nodes using the NS platform to simulate the chord protocol. At the same time, we agree on the following parameters:

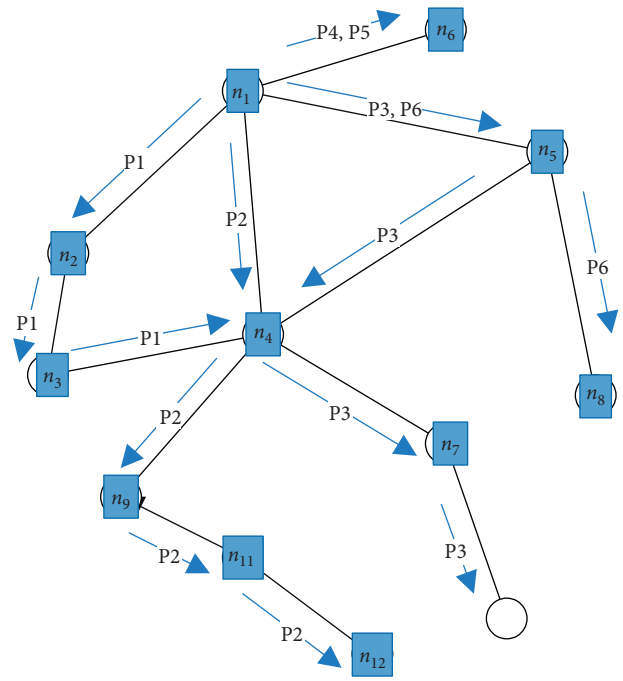

Figure 1: Query failed paths graph.

S: query success rate, the number of successful queries/the total number of pathfinder

mf: malicious node rate, malicious nodes/total nodes

ttl: query the maximum number of hops

mil: malicious node insertion rate. The number of malicious nodes inserted into the network per second

mdf: malicious node detection rate. The probability of detecting a malicious node

pl: the average path length of successful query

The first set of experiments is as follows:

In order to facilitate comparison and calculation, we suppose that 100 malicious nodes are inserted into the network per second, so *mil* is 100, the maximum number of hops of the query message is equal to 10, and we compare the comparison of query the success rate of *S* changes between chord and HFS-Routing. The average success query path length is *pl*.

The results from Figure 2 show that the query success rate of a and B was almost the same in the first 300 ms of the experiment. However, with the gradual increase of malicious nodes, the proportion of malicious nodes in the network increased significantly, and the success rate of chord query began to decline, while the query success rate of B was only slightly affected, so HFS-Routing can improve the success rate of queries than chord when the proportion of malicious nodes rises to a certain threshold. In the actual network, various attacks are increasingly popular, especially the prevalence of botnets, which leads to the continuous increase of malicious nodes, and HFS-Routing can better solve this problem and effectively defend attacks by malicious node.

Next, we will assess the impact of the increasing number of malicious nodes on the length of successful query paths to compare the advantages of our model. From Figure 3, by the

```
/ * *
 * DFS core pseudo code
@param n is the node from which the search is currently started
 * * /
bool FPD-Detect (Node n) {
      if (isEnd (n)) {//return true, once the search has reached an end state
         return true;
}
if (n.indgree > k1 and n.outdree = 0) {//return true once one is found. Return "n is a malicious node";
}
if (n. Edge > stack2.top.edge) {//Once it finds a node that diverges the path to be forwarded, returns that node
         return "stack2.top is a malicious node";
}
stack2.top ++ = n.edge; //Save the edge with n as the vertex
arr [n] ++;//record n how many access path;
if (arr [n]> k2) return "Malicious node"; //Back to the path through the node more than the number of trails, the path
//The first non-source node is a malicious node.
   for (Node nextNode of n) {//Traverses n adjacent nodes next Node
      if (! visit [nextNode]) {//
         visit [nextNode] = true; //Next Node cannot appear again in the next search
         FPD-Detect (nextNode)
         visit [nextNode] = false; //Set to false again, because it may appear in the next/search another path
      }
      }
   }
```

Algorithm 1: Detection algorithm of HFS-Routing.

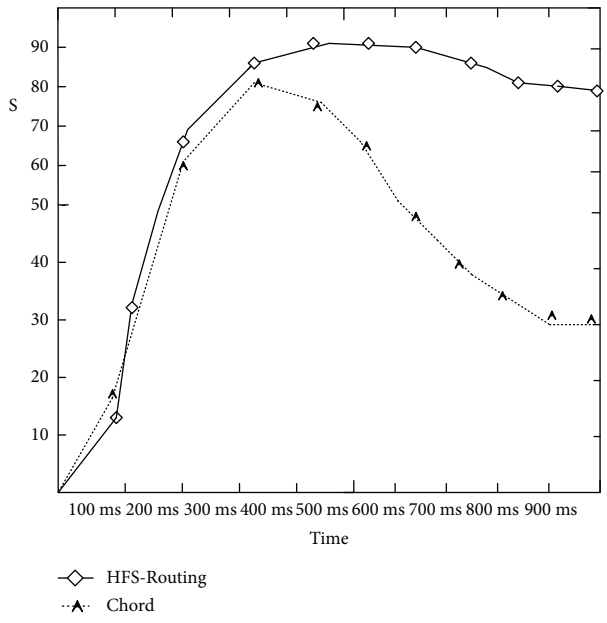

Figure 2: Dynamic contrast graph of query success rate.

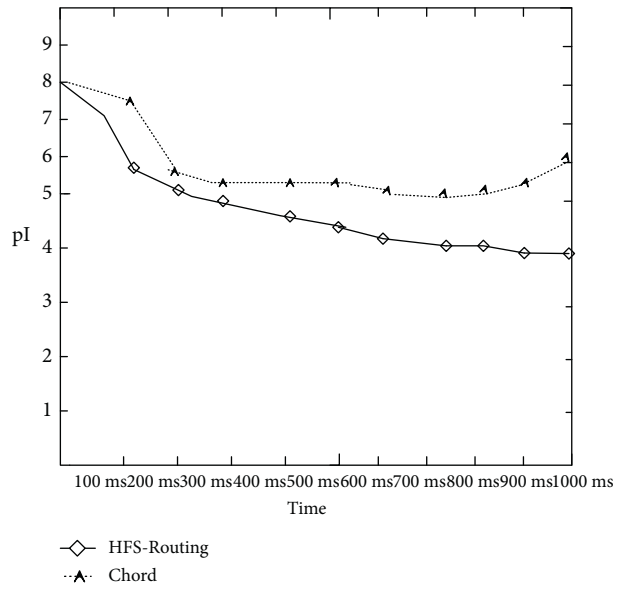

Figure 3: The average path length of successful query.

analysis of the average path length of successful query, with the influence of malicious nodes, the chord protocol will eventually lead to the deterioration of the routing performance and the big of the average successful query-path length of HFS-Routing has not changed significantly with the increase of malicious nodes.

Query can eventually be reduced by a certain value. Finally, HFS-Routing obtains a shorter query path length. Under the condition of successful query, the longer the query path length, it indicates that the query is forwarded to the malicious node in the middle of query path. In the case of redundancy, although the query is finally successful, it indicates that the efficiency of the query is reduced, so HFS-Routing can improve query efficiency and reduces query overhead, so it is more feasible.

4.1. The Second Set of Experiments. In order to compare the performance of the detection rate on malicious nodes with Symmetric-Chord [6] proposed in Luo et al. and SybilLimit [14] proposed in Yu et al., we will carry out the following

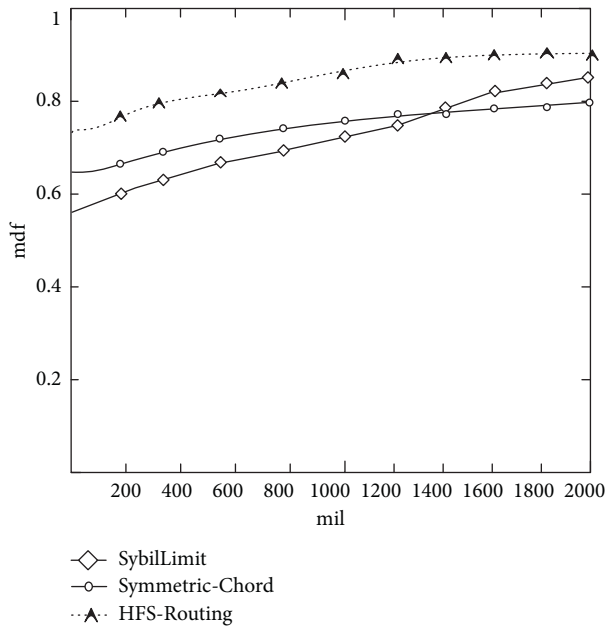

Figure 4: Contrast graph of detection rate on the malicious node.

comparative experiments, in which the maximum number of query hops ttl has effect on the detection rate on malicious nodes. Many experiments show that if *ttl* is small, then the query success rate of the algorithm will be reduced; however, when the *ttl* is greater than 10, the detection rate of the algorithm for malicious nodes will basically not change. At the beginning of the experiment, we set to query the maximum number of hops *ttl* is 10. Then, we gradually increase the malicious node insertion rate mil at the same time. The final result is shown in Figure 4 (The probability of detecting a malicious node).

From Figure 4, the results show that HFS-Routing has more advantages than Symmetric-Chord and SybilLimit in detection rate on malicious nodes if mil below 1400. When mil is above 1400, the advantage is not obvious. Considering that when most of the nodes in the network are malicious nodes, the detection rate of malicious nodes of all algorithms will increase so HFS-Routing has a higher detection rate of malicious nodes than other algorithms in practice. On the other hand, it does not need to sacrifice network scalability, so it is feasible.

5. Conclusions

In DHT-based peer-to-peer networks, malicious nodes attack on the routing through sybil attacks, eclipse attacks, and routing table poisoning. Most of the current researches on security routing realize safety at the expense of network performance or loss of feasibility, so we establish the model of the routing process. The common essence of the routing attacks is found to be the destruction of the convergence of the routing path. By the convergence detection, we propose a kind of secure routing mechanism HFS-Routing. Finally, the experiments are designed and analyzed. The results show that HFS-Routing has lower overhead, better scalability, and higher detection rate for malicious nodes. Therefore, HFS-Routing is a highly feasible security routing mechanism. The contribution of this paper lies in (1) establishing the routing model and (2) proposing security mechanism HFS-Routing, which can effectively improve the safety of routing. In future research, HFS-Routing will further improve the false positive rate of detection.

Data Availability

The raw/processed data required to reproduce these findings cannot be shared at this time as the data also forms part of an ongoing study.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported by the (1) 2019 Training plan for young backbone teachers in Colleges and universities in Henan Province: Research on Intelligent Decision Model of Internet of Things Industry Based on Knowledge Map (2019GGJS222); (2) 2019 Funding plan for Key Scientific Research Projects of Colleges and Universities in Henan Province: A Management and Control Model of Intelligent Agricultural Equipment Based on Deep Learning (19A520010); (3) Research Project on curriculum reform of 2021 teacher education in Henan Province: The Improvement of Normal School Students' Intelligence Education Literacy under the Background of Intelligence Teaching (2022-JSJYYB-066).

References

[1] T. Fujii, Y. Ren, Y. Hori, and K. Sakurai, "Security Analysis for P2P Routing Protocols", International Conference on Availability," in *Proceedings of the 2009 International Conference on Availability, Reliability and Security*, pp. 899–904, Fukuoka, Japan, March 2009.

[2] H. Ismail, D. Germanus, and N. Suri, "P2p routing table poisoning: a quorum-based sanitizing approach," *Computers & Security*, vol. 65, no. 283-299, pp. 283–299, 2017.

[3] F. A. Eichert, M. Monhof, and K. Graffi, "The Impact of Routing Attacks on Pastry-Based P2P Online Social Networks," in *Proceedings of the 2014 Euro Property : Parallel Processing Workshops*, pp. 347–358, Porto, Portugal, August 2014.

[4] G. Jaideep and B. P. Battula, "Survey on the present state-of-the-art of P2P networks, their security issues and counter measures," *International Journal of Applied Engineering Research*, vol. 11, no. 1, p. 616, 2016.

[5] C. Medina-López, L. G. Casado, V. González-Ruiz, and Y. Qiao, "An SDN approach to detect targeted attacks in P2P fully connected overlays," *International Journal of Information Security*, vol. 20, no. 5, 2020.

[6] B. Luo, Y. Jin, S. Luo, and Z. Sun, "A symmetric lookup-based secure p2p routing algorithm," *Ksii Transactions on Internet & Information Systems*, vol. 10, no. 5, pp. 2203–2217, 2016.

[7] H. Ismail, D. Germanus, and &N. Suri, "Malicious Peers Eviction for P2P Overlays," in *Procedings of the 2016 IEEE*

Conference on Communications and Network Security (CNS), pp. 216–224, Philadelphia PA USA, October 2016.

[8] H. Shen and C. Z. Xu, "Elastic routing table with provable performance for congestion control in DHT networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 21, no. 2, pp. 242–256, 2010.

[9] X. Xu, "Providing Efficient Secure DHTs Routing," in *Proceedings of the 2009 International Conference on Computational Intelligence and Security*, pp. 510–514, Beijing, China, December 2009.

[10] Y. Han, K. Koyanagi, T. Tsuchiya, T. Miyosawa, and H. Hirose, "A trust-based routing strategy in structured P2P overlay networks," in *Proceedings of The International Conference on Information Networking 2013 (ICOIN)*, pp. 77–82, Bangkok, Thailand, January 2013.

[11] T. Cholez, I. Chrisment, O. Festor, and G. Doyen, "Detection and Mitigation of Localized Attacks in a Widely Deployed P2P Network," *Peer-to-Peer Networking and Applications*, vol. 6, no. 2, pp. 155–174, 2013.

[12] C. Lu, X. Miao, and Z. Liu, "A safety algorithm of p2p routing based on multiple-encryption detecting technology," *TELKOMNIKA Indonesian Journal of Electrical Engineering*, vol. 11, no. 10, pp. 5815–5823, 2013.

[13] A. Castaneda, D. Dolev, and A. Trehan, "Compact Routing Messages in Self-Healing Trees," in *Proceedings of the 17th International Conference on Distributed Computing and Networking*, pp. 1–10, Singapore Singapore, January 2016.

[14] H. Yu, P. B. Gibbons, M. Kaminsky, and F. Xiao, "SybilLimit: a near-optimal social network defense against sybil attacks," *IEEE/ACM Transactions on Networking*, vol. 18, no. 3, pp. 885–898, 2010.